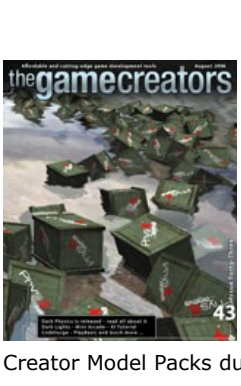
thegamecreators
Newsletter - Issue 43 - August 2006

editorial / august 06



Has the extreme heat melted away your game development energies? Need something to get revitalized? How about the final fully Ageia approved release of Dark Physics for DarkBASIC Professional? :) Yes folks it is finally here! After months, nay what seems like *years* of development, we have been given the green light to unleash the awesome PhysX power to you! and boy, it is some serious power you're getting. See the full article this month for the low down.

In other news we've got exclusive screen shots of two brand new FPS Creator Model Packs due out soon, and a new trial version of DBPro is released. There are 8 classic arcade games to play, PlayBasic news, the release of Dark Lights, a free Palette editor for Pro Motion, 3 superb tutorials covering AI, physics bridges and Demo Scene effects, and plenty more.

You'll notice a review of the book 'Freax Volume 1'. I treated myself to this book a month ago, and have been dying to tell you all about it. It covers the sub-culture of the computer demo scene, starting with the Commodore 64 and moving to the Commodore Amiga (Volume 2 will cover the PC/ST). It starts right from the scene origins, up to the tail end of 2005, and covers all of the major demo techniques - who created them first, and who bettered them. It is a truly fascinating book, and an inspirational tale.

Don't forget you can read this newsletter online, which is a much nicer experience if you are stuck within a web mail client. See you next month for issue 44, but for now I'm off to watch a few more classic Amiga demos (anyone up for a TGC demo compo?!)

Cheers — Rich

contents

(1) Dark Physics is here! Plus what happens next... (2) Dark Lights Released (3) 2 New FPSC Model Packs (4) CodeSurge IDE (5) Mini Arcade (6) Free Palette Editor (7) PlayBasic News (8) DBPro Mastery (AI Evolved) (9) From the Forums (10) DBPro 6.2 Trial (11) Lee's Tip - Bridges (12) Freax Vol. 1 (13) DBPro Tech Tips (Demo Scene Style Effects) (14) Forum Banner Winners (15) New Matrix1Utils (16) Synergy IDE Update (17) The Winner (18) Outro

Dark Physics has landed + What are TGC doing next ...

It's here! After months of development, doc writing, example coding, testing and revisions - the brand new Ageia PhysX™ approved Dark Physics has arrived! For those of you who've pre-ordered already, jump directly into your TGC Order History and grab the downloads; it is now ready and waiting for you! You will find two downloads available. The first is the main installer (62MB), this will install the latest PhysX™ drivers, the Dark Physics DLLs, some examples, and all the help files. The second download is Dark Physics Projects (54MB) which contains all the rest of the example projects and media, bringing the total up to 115 unique examples to load, run, rip apart and learn from.

We have also launched the new Dark Physics web site. Hop on over to http://www.thegamecreators.com/?m=view_product&id=2025 and take a look! You'll find 6 videos of Dark Physics in action, as well as 3 stand-alone demo programs to run.

On our forum we have created a new Dark Physics board, specifically for discussion about Dark Physics. Feel free to drop in, post a message, talk to the developers, show us your code and learn from fellow devs!

We're going to be holding a little Dark Physics contest in the forum in a few days time, so be sure to take a look!

All that remains is for us to say a big thank you, both to Aegia for their superb support during the long development process (and for creating kick-ass physics cards of course!), and to the hundreds of you who pre-ordered and have been eagerly awaiting this release. We hope you like what we've done, and we can't wait to hear your feedback (good and bad!) on the forum.

Dark Physics is on sale NOW priced $59.99 (€44.99, £29.99)

So what are The Game Creators working on now?

Here is a sneak peek look at what TGC will be bringing you in the coming days and months. First up is a brand new update to the **Dark Game Development Kit** (DarkGDK), the new name for the DGSDK! This will bring the DGDK bang up to date, in-line with version 6.2 of DarkBASIC Professional. This work has been finished, and we were hoping to be able to release it today with this newsletter, but some final tests were required first. Even so, you can expect to see it VERY soon! Lee Bamber, lead developer, explains:

"After many weeks working on a whole slew of function and help fixes in the DarkGDK, formerly Dark Game SDK, we are now very close to the release of version 1.6.2. As requested by the DarkGDK community, we have dropped support for Visual Studio 6, and added support for Visual Studio 2005, which means you will be able to use DarkGDK with the free Microsoft(r) version of Visual Studio C++ Express. We have also extensively tested all the sample projects to ensure your introduction to the library is as effortless as possible. Finally, thanks to a dedicated DBPro/DarkGDK user, we have made extensive corrections to align the function names to the DBPro equivilants, so if you know the command in DBPro, you can work out the command easily in DarkGDK."

Pretty exciting stuff if you've been waiting for a DGSDK update for a long time :) But what else are we working on? Well, fresh from his sterling work on Dark A.I. and Dark Lights, Paul has been slaving away on **Dark Shader** (working title). This is a visual shader development system, using the brand new Windows Vista look-and-feel interface (but it will run fine on XP). It allows for visual creation and editing of shaders. By simply dragging nodes from one effects panel to the next you can build up impressive effects in real-time. The shaders can then be exported directly to DarkBASIC Professional. We will bring you more news on this as we have it, but Dark Shader is currently scheduled for an October 2006 release date.

We have released a new version of **Dark A.I.** today. Version 1.03 implements a few change requests from the community, and fixes some minor issues. You can download this new version right now from your Order History.

On the DarkBASIC Professional front we've got the new help files to finish and release, as well as a few other nifty changes under the hood. Trust me, this is a GOOD time to be a DBPro user :)

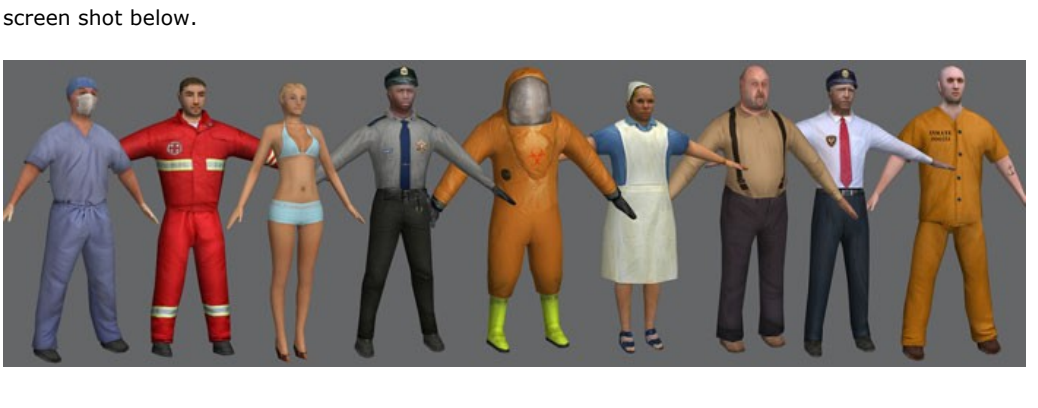
Dark Lights is Released

We gave you a quite in-depth look at the Dark Lights plugin for DarkBASIC Professional last month. However we wanted to mention two things: First of all, this plugin is now fully available to purchase! It costs only $25, yet delivers a powerful light mapping framework that you can integrate directly into your own games and tools.

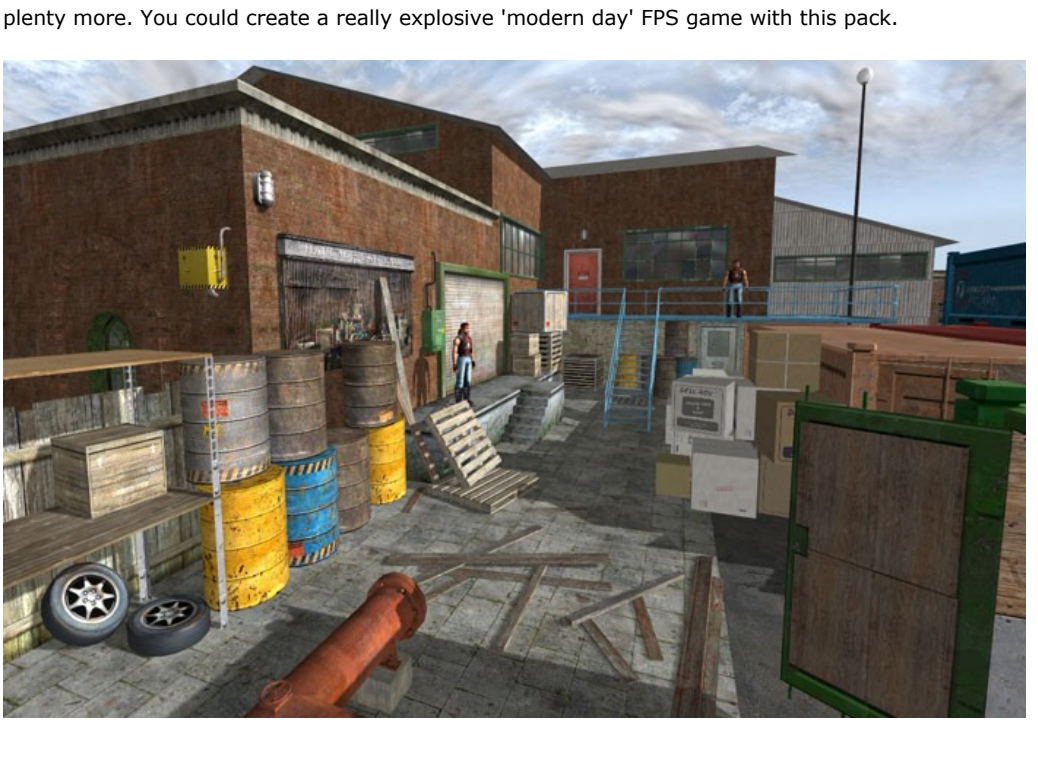
We have also released a demo version of the Dark Lights editor. The editor allows you to place lights, set their colour, range and style, and light map any DBO object. This editor is just an example of what you can make with the Dark Lights DLL, and the full source code is included when you buy Dark Lights. Please note that saving DBO files on the demo version misses out random polygons from the object. You can download the demo and read more on the Dark Lights web page.

2 New FPS Creator Model Packs

We are pleased to announce that two brand new FPS Creator Model Packs are nearing completion, and are due for release shortly. The first pack concentrates on providing 20 new characters for your games. These are fully textured, fully animated characters. You can easily tailor their clothing textures for variety, and the models include characters such as: A prison inmate, a bio-hazard suit, a Doctor, a super-model, and more. Different textures provided with the characters will give you an increased amount of options (the 'farmer' guy can become a Zombie for example). So there should be plenty to keep your games occupied. A few of the characters can be seen in the screen shot below.



The next model pack is called 'The Warehouse'. As you can see from the screen shot below this includes all kinds of Warehouse building segments and objects to populate your scene with: including packing boxes / crates, pipes, gantries, walk-ways, tyres, barrels, lights, alarms and plenty more. You could create a really explosive 'modern day' FPS game with this pack.



Both packs should be available in the next couple of months, and will retail for a similar price to the Model Packs already on sale on the FPS Creator web site.

German FPS Creator Website

On the new German FPS Creator site you'll find everything FPSC related from demos, models, FAQs, product chat, screenshot gallerys and more. There's also the 30 day trial demo available.

Visit the German FPS Creator web site at: http://www.fpscreator.de

CodeSurge IDE for DBPro

CodeSurge is a brand new IDE by Hyrum Richter for DarkBASIC Professional. Although still in development it already has a strong forum following, and is declared as being one of the better alternative IDEs out there. Fast, efficient and professional, it features a good array of IDE functions including:

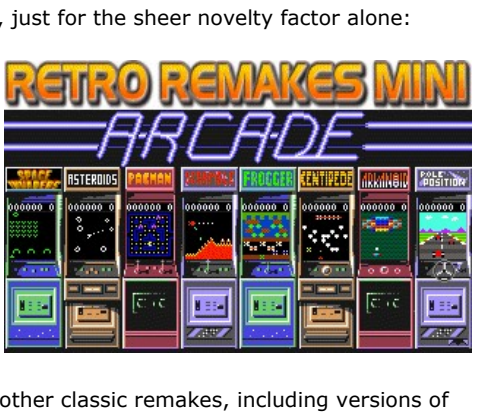
- All the functionality you'd expect from any edior (cut, copy, paste, find, etc.)
- Syntax highlighting for third party commands as well as the built in DBP commands.
- IDE jumps to the correct line an error occurs on as long as the compiler reports it properly.
- Tabbed interface, with the option to cascade or tile your IDE windows in an MDI layout
- Customizeable layout with Visual Studio style dockable forms.
- Code explorer lists all your functions, labels, and includes.
- Very small, fast, and stable executable. Download size is under 1mb and there is no noticeable slowdown on even very large projects.
- No one has ever reported that it ate their code!
- Option to associate DBP files with the IDE.

And the best feature of all is that it's free :) Just as we were putting the final touches to the newsletter, Hyrum emailed to say that he has added a great new feature: "Well, I just added perhaps the most wished for feature in all of DBP history :) highlighting of RUNTIME errors. This means if in the middle of running your game, you get an "error, cannot load image at line 1234", the IDE will jump to the correct line, even if it's in an include file." - Nice one!

You can download the latest version from the CodeSurge forum thread.

The Minionsoft Mini Arcade

We've been big fans of Minionsoft for many years now. They produce great arcade game remakes using DarkBASIC Classic, and have churned out a consistently impressive collection of playable and addictive games. This one really caught our attention, just for the sheer novelty factor alone:



It's a complete **Mini Arcade**. You get 8 arcade machines to pick from, and can play all of the games on offer - they run within a tiny arcade cabinet, the displays being only 32x32 pixels in size! Yet you'd be amazed how much detail they have packed in. Full recreations of all the classics are there: Space Invaders, Asteroids, Pacman, Scramble, Frogger, Centipede, Arkanoid and Pole Position. My personal favourite is Arkanoid - what's yours? Grab the download and enjoy. You should also check out the rest of the Minionsoft web site for other classic remakes, including versions of Uridium, Atomino, Atic Atac, Repton and loads more.

Visit http://www.minionsoft.com. Download Mini Arcade from our file server mirror (1MB)

Free Pro Motion Palette Editor

Drawing palette based images and animations can be painful when it comes to manipulating or rearranging colour palette entries. Often you want to copy move or swap a couple of entries or you need to sort a selected range of colours. The built in functions of Pro Motion are quite limited when it comes to this. That's why there is a new freeware plugin that helps you to easily rearrange and organize your palette colours.

Take a look at the Palette Editor page. There you will find a quick view of the new plugin, and you can download the installer. You will need Pro Motion V5.1 (full or trial version) to run this plugin.

Pro Motion is a pixel perfect art package similar to Deluxe Paint, and can be bought from The Game Creators website.

PlayBasic News by Kevin Picone

PlayBasic - Ready For the Next Big Step

After a few months of focusing entirely upon command set embellishments, this month our attention has firmly shifted back to the bigger picture. That of growing our core technologies to the next generation and beyond. Central to this growth is the expansion of the PlayBasic compiler and runtime environment, the VM (Virtual machine). It might come as a surprise to some readers, but our objectives behind PlayBasic, have never been to just produce a feature rich 2D programming language, but rather, to build a core language that would be central to a family of specific programming tools. Ranging from the 2D to 3D, or even scripting. Where the concept is to allow programmers to use the same familiar foundation with a user selectable engine.

VM2 - The engine that drives the machine

Over recent months we've been talking about and implementing some of necessary internal changes to embark upon VM2 development. Which is now fully underway. PlayBasic works a lot like Java / Visual Basic and even DB classic. Were the compiler turns your source code into a property low level language. This language is not machine native (ie. machine code) but rather a virtual machine byte code. Execution of this compiled code is performed by a emulator of sorts called a virtual machine.

You are no doubt wondering why stick with a VM2 and not compile directly to machine code? Firstly, languages that build machine code exe's generally don't do it in one step. The reason is not that it's incredibly difficult, in fact there's hardly any difference, but rather dumping machine code directly, not only makes optimizing clumsy, plus the resulting code is CPU dependant! Compiling to an intermediate language first, allows pre-optimization to occur. This allows program logic to be further streamlined (the number of operations reduced) resulting in faster code! This can be demonstrated in the PB v1.46 patch (below), were we've been testing a simple post optimization mode which can reduce the number of operations used in math calculations. Even with some 'simple' opt's the performance difference of our test code improved by around 25% / 30%. Number Crunching comparison between PB1.46, DB and Dbpro.

The compiler / VM design is incredibly flexible one. Allowing us to build only one compiler, but as many variations of the Virtual machine required. So we could implement JIT (full or partial compilation to machine code), translate the byte code to Assembly / C perhaps, make a standalone versions as scripting plug in's (how about using PB as a DBPro scripting language?) or what about running your PB programs on other platforms? ... It's all possible!

Touching on PlayBasic FX, a PlayBasic 3D?

Earlier we mentioned the next generations of PlayBasic. While you might assume is that these successor products will mean the death of PlayBasic. This **couldn't** be further from the truth. In fact, what's occurring is that PlayBasic is transforming into a more modular system. Where modules can be added, remove or replaced. This modularization will allow users to use the one base development environment and selectively choose the type of project you wish create. To make a crude DB analogy, it'd be like writing DB project and being able to selectively compile your project with DB classic or DBpro. Thus PB users will be able to write exclusively 2D or 3D projects from the one tool!

What's *PlayBasic FX*? PBFX is what we call the DX9 (hardware accelerated) equivalent of PB. Like PB, it's primarily focuses is 2D. The difference is really in the behind the scenes implementation. This will allow users to choose what level of requirements (DirectX 3 or DX9/Dx10) your programs have. So PBFX programs can use the latest hardware acceleration, while PB programs take the lower requirement approach, giving the best of both worlds! Read the PlayBasic FX FAQ for more details.

PlayBasic 3D? PB3D is obviously PB's step into the world of 3D. While we'll be providing our own 3D engine. The real treat is that due to PB's up coming modular design, users will be able to use 3D party engines with PB also. For example, DBPro owners could potentially use DBPro's engine with PB. **Reminder:** existing PlayBasic users will get the PlayBasic FX module FREE!

From the PlayBasic Forums

This month we've seen a few particular innovative releases from PB users, so lets check them out.

Retro Racers (isometric racing game)
Early Look at Darco's isometric RPG
Shadow Of The Beast Remake (src)
BOTS - Brotherhood Of The Solarbeetles (src)

Updates PB1.46

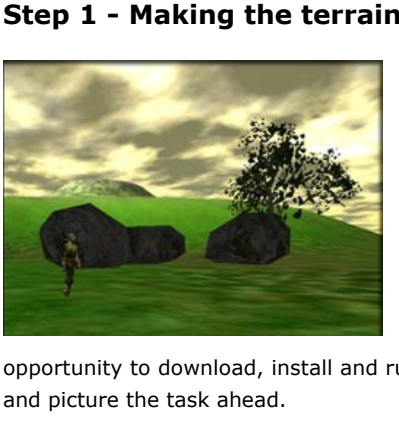
Well until next month, I'll leave you with the latest update retail update PB1.46, Have fun!
Download Latest Retail Patches.

DBPro Tutorial - Making AI more intelligent - by Steve Vink

As a young tearaway, I was always in trouble. My main problem was the instruction "Whatever you do, don't go <insert dangerous place here>". Even though I'm the owner of at least a dozen X-rays of various limbs and broken bones, I still think it was healthy to be inquisitive at that age. My inquisitive nature got the better of me when I first started playing with the Dark AI plugin. And to my delight, I discovered I could change the intelligence of the system to my advantage. Just like my own parents, I found that it's possible to say "Don't go there!" whilst still allowing my characters to venture into dangerous territory when it's absolutely necessary, or just too exciting to resist.

In this tutorial, we'll use a realistic example. We'll set up a scene with a small number of obstacles, and a hill between 2 camps. The challenge will be to persuade the character to take the easy route and walk around the hill, but at the same time allow him to climb the hill in times of peril or attack.

Step 1 - Making the terrain



Firstly, we need a setting. This example uses Advanced Terrain to create the ground, which is part of the standard DarkBASIC Professional command set. It generates a large hill in the centre, and also encloses the playing area with a raised area, a useful technique for removing unsightly edges and implying that the land is larger than it actually is. I have used TreeMagik and Plant Life to add a small number of obstacles, including the rocks that define the 2 camps. This is carried out in function makeTerrain(). Take this opportunity to download, install and run Step 1. You can use the arrow keys to fly over the terrain and picture the task ahead.

At this stage, the first steps of the AI have also been introduced:

- The rocks have been added directly to the AI system as obstacles.
- The bases of the trees have been added by placing a cube in the same position and using this as the obstacle template. We cannot use the tree directly, because Dark AI would assume that the overhead foliage was part of the obstructed area.
- Several other waypoints have also been manually added at the top of the hill and around the base (function createWaypoints()). This is to ensure that the character we introduce later has various options available to him.

Step 2 - Adding the Character

Now it's time to see how a character reacts to our newly created environment. This part is ridiculously simple. After loading the character and setting a few basic parameters, he is added to the AI system (function makeCharacter()). A path is created between the start and destination, and the character is instructed to use this path as his Patrol. You can compile and run Step 2 now, and watch what happens after implementing just a few lines of code.

We have some success in that our soldier is walking around under the control of the AI system. However, things are a little strange to say the least. Firstly, the soldier has no interest in the intermediate waypoints, you will notice that they are ignored. Secondly, he has taken a liking to the trees in the scene, and takes quite a detour on his way to the destination. So what is happening?

The soldiers preferred path between 2 points is a straight line. However, the rocks in the scene prevent this. After looking around for an alternative path, the soldier determines that the best waypoints to use are those that define the tree border. Despite the presence of a large hill and intermediate waypoints, the AI system allows the soldier to "see" the tree and use this as the best option.

The soldier is currently achieving his aim and reaching the second camp, but the situation is not ideal. We need to reduce his range of sight, give him more options and finally persuade him to walk around the hill. You will have noticed that the code causes the soldier to slow down when ascending hills. In reality, it would be natural for him to walk around the hill if it was the faster, easier option.

Step 3 - More waypoints, more choices

In the next step, we will add more waypoints, reduce the "look-ahead" ability, and the result should be closer to our desired effects. Firstly, adding more waypoints is a simple way to add more options for our characters. Waypoints are created automatically by obstacles, but in scenarios like ours they are few and far between. In extreme cases, the entities can actually suffer from agoraphobia and refuse to move from their position. Taking performance into consideration, several waypoints are added in this version of the program.

Waypoints have been added roughly every 100 units, with a random factor to prevent clinical straight-line paths. After adding them, updating the visibility is essential, and this is also where performance is carefully controlled. Here is the command as it is used in our tutorial:

```
AI update waypoint visibility 0, 120
```

The final parameter dictates the maximum distance of the paths generated between points. This has been set so that adjacent waypoints will form paths, but those further afield are out of range. Without this parameter, every point would be joined to all others by a rats' nest of paths. This would add a huge burden to the AI system, and be an unnecessary complication in our scenario. At this stage, we have also added a similar parameter to the path-creating command:

```
AI make path between points 10, 0, x1#, z1#, x2#, z2#, 300
```

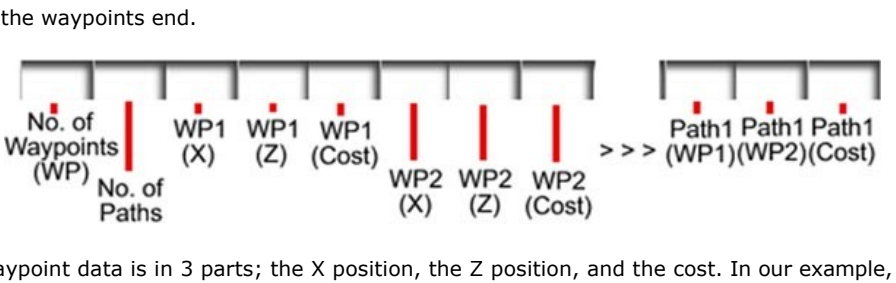
The final parameter in this command ensures that a path can only be constructed from waypoints within 300 units of the previous one. This will stop our friend from seeing the tree in the distance and using it as part of his path. He must include more waypoints in the path to find his way from one end to the other. Compile and run the code again, and check the results.

Step 4 - The finishing touches

The results of the previous stage speak for themselves. Our soldier can successfully find his way from A to B without taking obscure detours. The final step is to ensure he doesn't climb that rather large, steep hill in the centre which slows him down so much. The obvious solution is to make the hill an obstacle. Then he is forced to find an alternative route because it is impassable. But then it's also out of bounds permanently. You can't flee up the hill, attack up the hill, or mount an unexpected assault on your enemy from above. You have effectively built a wall around a large section of your landscape.

Dark AI has a fantastic feature that we can take advantage of to accomplish this task. The waypoints and paths can be copied into a memblock for us to manipulate. First, it's important to understand how pathfinding works. In it's simplest form, the AI module looks at all of the possible paths and works out which is the shortest. It does this by looking at the 'cost' of each edge between waypoints, which is actually it's distance. The waypoints themselves also have a cost, although this is zero by default.

In the tutorial source, the waypoint and path data has been copied into a memblock (updateWaypoints()). The waypoints occupy 12 bytes each, from position 8 in the memblock. The float value at Position 0 tells us how many waypoints there are. Incidentally, position 4 indicates the number of paths, and they also occupy 12 bytes each, continuing in the memblock at the point where the waypoints end.



The waypoint data is in 3 parts; the X position, the Z position, and the cost. In our example, any waypoint in a position on the map that is higher than 60 units has been modified. The cost of travelling through the waypoint has been updated to 2000, a value way above the potential cost of walking around the hill. This is starting to sound complicated, but I assure you it isn't. Let's look at the pseudo-code, which explains the source code that you can also view in the downloaded files:

- Load waypoints into memblock
- Get waypoint count (Position 0)
- For each waypoint in memblock
-
    - Get X and Z positions
    - Get ground height from map
    - If ground > 60, add 2000 to waypoint cost
- Make waypoints from memblock

Compile and run the last example; it's time to see if the not-so-hard work has paid off.

The final word, and an important little trick - Our soldier now makes the intelligent decision to walk around the hill. But at the same time, the hill is still available to hide in, attack from and run away from the enemy. In fact, because the return path of a patrol is directly to the start point, our soldier will return back over the hill to prove that it is still accessible.

One thing to notice is that after updating the cost of the waypoints, we did not need to update the waypoint visibility. Why not? Well, it's because all of the waypoint edges are still in tact, it's only the cost of passing through them that has changed. And why is this an important little trick? It means that you can interactively change the way your characters move around the map without the overhead of rebuilding the visibility, which is a costly exercise. For example, you could prevent an entity from returning down a corridor by changing the cost of the waypoint directly behind him. You could burn down a building, and make the paths leading to it too costly to consider. My favourite use of this technique is to randomly increase the cost of waypoints in an arena to make the AI system less predictable, and ensure every game is different.

Until next time, Happy Coding!

Download this tutorial and code.

From the Forums

Here is a selection of current hot topics worth checking out ...

2D Graphics Buffet

David Gervais has released a stack of free 2D artwork you can use in your games. It includes graphics suitable for a breakout game, slot machine, tetris, mahjong, asteroids, cards, millipede, chess, pipe mania and loads more! Excellent stuff.

3D Graphics Buffet

Not content with releasing great 2D artwork, David has also released a load of 3D models too. Crates, barrels, poker chips, buildings and more can all be found in this thread. Again, you can use it all royalty free.

Leadwerks Engine Demo

3D World Studio author JoshK has released a demo of his new engines portal occlusion system.

35 Degrees

Fallout has released a new tune, well worth checking out in these balmy summer days. Sit back and chill.

Enhanced Audio Plugin

Torrey releases a new beta of his Audio DLL for DBPro. This provides a large amount of new audio related commands to both analyse and play all kinds of audio (CD, MIDI, Wav) along with extracting buffer lengths, bit rates, pitch shifting and more.

DBPro can get you a degree!

A bit of a novelty entry, but interesting none-the-less! Fallouts final year University dissertation was on using DBPro to make an RTS style game. It got him 73% (a 1st class top level grade) incase you were wondering :)

DarkBASIC Professional 6.2 Trial

It has been a *long* time since we updated the trial version of DarkBASIC Professional, but we have now released a brand new build based on Upgrade 6.2. It comes with a new installer, more example programs and tutorials, and offers the usual 30 day evaluation. It allows those of you who have never tried DarkBASIC Professional before to use the latest version and run the latest code, such as the excellent 'Scene Effects' given away in this months newsletter.

The trial version can be downloaded from the main TGC web site and is now the 'standard' version.

Download the 6.2 DarkBASIC Professional Trial.

Lee's Tip of the Month - Your Game is like an Onion

No, it is not that your game stinks and makes people cry, but just like an onion, your masterpiece is made out of layers. The layer I want to peel back this month is the very last layer of your game, and the one everybody will experience when they play your game for the first time. What our industry affectionately calls 'eyecandy'. Once upon a time, a fancy title page was all you needed to show off how cool your game looked. These days, although ideas still stream from artists, the buck now stops with the coder who has to make the graphics 'do something'.

The response you want from your audience is 'wow, that's a really cool bridge, it looks so real'. Of course if they have just finished playing the latest greatest triple-A game, you're going to have to do something pretty amazing to get that quote. Fortunately, we here at TGC are averse to sleep and like nothing better than to provide 'pretty amazing' in bite sized chunks.

To that end, we have produced a small demo in DBPro and Dark Physics, demonstrating a pretty amazing bridge for the next game you make that needs one. An mpeg is available for those who don't have Dark Physics (and cannot compile the source). To impress gamers of today, you not only need to make it look real, but feel real. And to make a bridge feel real, you have to make a real bridge. That means ropes, planks of wood and end posts. You tie the pair of ropes to the posts at each end of the gourge and lay the planks on the ropes. You tighten the ropes so it is strong enough to allow a character to walk across. Presto, your game has a bridge that can't look any more like a bridge, because it is a bridge, in every respect.

The exciting thing about our new appetite for eyecandy, is that we have now entered a realm where the attention to detail has given us some exciting benefits for our game designs. We have a functional bridge, which means we can interact with our game bridge using rules we have learned from real life, instead of a user manual. For example, we know that if we cut the rope, the bridge will collapse and no enemies can follow that way. We know that the bridge has wooden planks, so we can pick up one of the planks if we needed firewood elsewhere in the game. I have been writing games for almost as long as they have existed, and even I cannot fully grasp the potential of what this means for the future of games. Who knows, maybe someone reading this will find enlightenment and evolve a whole new kind of game? Exciting times!

Download the code and video (22MB)

Freax Demoscene History - Volume One

In a small bought of nostalgia last month, I featured the Amiga Demo Podcast. This month I bring you news about one of the most fascinating books I have read in a long time: Freax - The Brief History of the Computer Demoscene - Volume 1. Here is the low-down on this superb book:

"Freax: Volume 1 (ISBN 3981049403) is a fast-paced, comprehensive look at the amazing stories of the digital art community known as the "demoscene". For over 20 years, computer programmers, artists, and musicians have tweaked, optimized, and pushed computers to their limits... all for the sake of art.

Anticipating advances in graphics and visuals by years, the members of these computer scenes have changed the face of computing. Now, their story has been told. Within this 224 page book, a wide range of subcultures are examined, ranging from the early 8-bit beginnings of the 1980s to the present day world of gigahertz and gigabytes. Scenes such as the "warez scene", "hack / phreak", "underground ASCII", and many more make appearances, and key figures from these scenes are interviewed.

Years in the making, this book contains a comprehensive first look at the unique demoscenes of the world, accompanied with over 480 full-color screenshots and illustrations. Whether you were a member of the scene or just wish you were, Freax tells a story you'll want to hear."

"When we got our first computer in our childhood, none of us knew that we're going to change the world someday. Playing games and typing simple BASIC programs was intended to be fun. Swapping games with the guy next door was nothing serious. It still wasn't when we wrote our first contact letter to another guy in some other town; and finally we found ourselves in a dazzling network of contacts, great projects running around us, brilliant pieces of digital art filling our screens.

This was the magic of the demoscene.

Years have passed, and our innocent childhood hobby has grown into an amazing worldwide subculture. Democoders, crackers, importers, musicians and graphists from all around the world changed the way we think about computers and the world itself, changed our life, changed us forever. We are special, we are demosceners. No outsider can possibly understand us. We are FREAX."

Now I'm aware that a lot of you don't really know WHAT the demoscene is (or was), and that quite a number of you are too young to have ever owned a Commodore Amiga, let alone a Commodore 64, but if computer generated graphics and history fascinates you, I cannot recommend this book highly enough. The research and production values are extremely good, the screen shots both captivating and inspirational. For $36.99 you can't really go far wrong!

You can order Freax Volume 1 by PayPal directly from the web site http://www.freaxbook.com. Please note that if you're in Europe I would strongly recommending emailing the guys and paying an extra $9 to have your order upgraded to Air Mail. My copy of the book arrived in only 5 days (from the US to the UK) because of this, otherwise it could have taken up to 8 weeks (dreaded 'surface' mail). If you're in the US already, you've nothing to worry about.

You can also order it from Go64 (good if you live in Germany) or Amazon.com (but they charge the most!). Whatever route you take to get this book, make sure you do so, it is well worth it.

Tech Tips - 'Scene' Style Effects - by Grant Clark

Hi all, and welcome to another session of non reality, this months clinic will deal with seeing into the future... er.. DBP scene style effects made easy, and I'll introduce you to a collection of code snippets that will help you on your way to making cool intros and demos with DBP.

So what is the scene all about, and why is it so important? (Editors note: Check out the Freax book review this issue for more info!) Put simply, the 'scene' is a massive underground movement of coders artists and musicians who produce realtime art using all manner of computing technologies, from old 8 bit micros through consoles and handhelds and the latest cutting edge hardware devices. The sceners will hammer every ounce of power out of the hardware to produce some of the most amazing realtime audio visual entertainment that you will ever see and hear. Scene demos and intros are non interactive affairs as a general rule, although some do have interactive elements in the code, it's not something that's common.

The whole point is to show off the skills that you and your coding friends or group have. Finished scene demos and intros are usually first seen at massive conventions and competitions, the biggest of which is probably Assembly , a compo held in Finland every year to find the best scene demos and intros, the 2006 event kicks off on August 3rd. This is where the games industry gets most of its fantastic effects and ideas from, the stuff that sceners do. Indeed a vast number of game developers started out coding demos (and lots still do). But not all scene demos use cutting edge technology, anything that looks and sounds good is what we are after, after all it's possible to do some really cool stuff with old technology like vector graphics and alpha sprites. It's how you combine the elements and present the final thing thats important. DBP is more than capable of producing much of the stuff sceners are famous for quite easily, which is great for us.

DBP AND THE SCENE

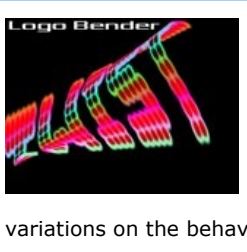
And whats the point of it all? The point of all this is to push *your* skills and artistic abilities, writing games is all very well and fine, but to make your games shine you need to polish them and thats what the scene is all about. Developing small scene style demos is a quick thing to do, much faster than making a game, after all there's very little or no interaction with a player, it is just wall to wall game style code thats heavy on the visual aspect. By learning how to make scene demos and getting a feel for what looks good where, you'll be developing your game writing skills on the fly. Virtually everything in scene demos is relevant to games, so it makes sense to make a few and to learn how all the fancy things are done for yourself, some of them are a lot less complex than you think, especially with the power and awesome command list you have using DBP. Scene demos are also a great way to get more code out there, turning experiments into cool intros and demos, the more code you have out there, the more you will be noticed. You'll gain kudos and respect from other coders as well as scene nuts who just love demos (** points at Rich **). Some people love a good scene demo or intro just as much as the latest and greatest games.

How do I get started making a scene demo or intro? Alrighty then, lets get down to business. Normally I would just write a commented scene style demo with an explanation of how it all works and fits together, leaving you with only a demo to mess with, but not this time. This time I decided to write a small bunch of popular scene style effects as seperate demos, you'll find 7 small demos each showing a different effect along with the source and media in the download for this article, each snippet is based on a set of functions you can use in your own demos, intros and games. The idea here is to take the different elements supplied and meld them together into something that is visually pleasing and interesting to watch. here's a list of the effects and what they do, but it's up to you to combine and rework them into something really special, a fun process full of surprises.

Blendy Blobs

This makes a lava-lamp style effect on the screen that appears behind a subtractive image, The blob effect here is faked using the brightness falloff of the texture to replace a lot of complex math. any image can be used as the subtractive part, it might look odd with some images though, best used with greyscales. feeding the subtractive image with another cameras view in realtime might be something interesting to try with this, or using a plasma as the subtractive texture could have spectacular results.

Logo Bender

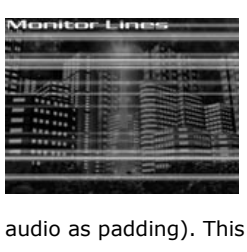


This effect uses sprites to rebuild a source image that can be warped and twisted to the extent that the sprites actually look 3d with thickness. This is all an illusion created using lines made from sprites to make curves, its a very nintendo style effect and would not look out of place on a nintendo 64 game title screen. This is a very cool effect although the function takes a few parameters to get lots of variations on the behaviour of the effect. I expect to see this added to a few title screens and stuff. sprites are fairly limited in their abilities, but used properly they can take on a whole new dimension.

Parallax Backdrop

A rather cool effect thats made using a whopping 3 objects, use the cursors on the demo screen to chance the scrolling direction and speed to get a feel for what you can do with parallax layers as backdrops to your games and demos. this effect would be rather cool as the backdrop for a scrolling shooter game as well as a demo or intro. Seamless textures that look good together are all you need to make nice parallax backdrops, an additional subtractive layer in this could be a very nice variation on the effect.

Monitor Lines



A very simple effect that simulates scanline interference like an old video tape, works well over monochrome or greyscale images to make them look like old video, this is what I done for the demo with this effect, placed it over a black and white scanlined image. It's also very handy on techy looking menu screens as extra filling on the screen without totally obscuring the image in general. (like white noise in audio as padding). This simple effect could be made better by adding smaller shorter lines that have random positions on the x and y axis.

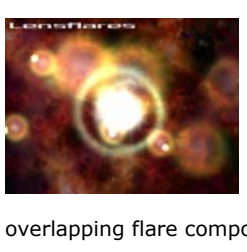
Zoom Starfield

This is a basic starfield simulation with flickering stars that rush toward the camera, a classic effect and an absolute must in virtually any scene style demo. Not too good an effect on its own on it's own, but combine it and you start to get more interesting results, starfields are very relaxing to watch, so no falling into a trance in front of the keyboard. Adding a sine wave to the stars z axis motion could result in the illusion of a jellyfish swimming into the screen with the correct textures and backdrop of course.

Drawn Starfield

Now this is more like it, It's a starfield same as the last but the stars are placed using a stack of sinewaves, the effect is like chasing a ribbon of fire through space. combine this with the Zoom Starfield and you'll be halfway to something really cool looking. This is one of my favourite effects, simple but very nice. Using a bit of symmetry and rendering the view to textures can make this a most amazing effect. using this inside a zooming tunnel environment would add much needed extra speed to the overall feel of the thing.

Lensflares



Based on code by VanB, this function will allow you to attach up to 50 lensflares to active DBP objects in your scene, very handy. The function is still a prototype but very usable. In the demo for this I have attached lensflares to 4 star like objects that follow strange orbits around an invisible gravity source using sine waves, this looks quite cool and produces some very unusual patterns in the overlapping flare components, I'm sure I saw a supernova in there at one point. I also quite like the idea of basing the flare visiblility on a the result of a raycast operation combined with a brightness and colour value assigned to the flare when it was attached to the object, could be very cool. The concept used in the lensfare code is simple enough to be reworked into a whole array of effects, just use a bit of imagination.

Each demo is based on a set of functions to control the effect, and has its own documentation in the source code. There's way too much stuff to have in the newsletter article text, so it was added to the code to make things easier. As you can see the code itself is nothing complex and the secret lies in the power of transparency and illusion more than anything else. Making functions like this is a good way to set yourself small realistic challenges, if you make a few of them you eventually end up with a small library of pre-written effects. Having such a library at your disposal will pay dividends when it comes to knocking out games that use the effects you developed as quick scene demo components. for instance the lensflares in this little lot could be used to add extra depth to explosions in a game by having the source of the explosion flare the camera view. Missiles whizzing past the screen can have a flare attached to the engine nozzle adding instant eye candy, or you could just use them as I have and make a cool backdrop effect for the sake of it or for in game menus etc. Write your code as small function snippets that are little scene demos and get more from the work you put in.

When it comes to making that killer game you'll have a whole host of nice effects just waiting to make your creation shine. With the approval of Dark Physics granted, it's going to get very interesting in the code snippit arena as people start to use physics simulation and trickery to power effects for more variation and realistic interactions in the scene. The addition of darklights and extends to the affray takes the possibilities for super cool intros and demos within DBP to new heights and with more speed and features too. Dark AI could proove interesting in giving demos a sense of autonomous behaviour with purpose. Anyone can make a scene style demo, so fill an empty screen with something nice today. If you want more info on the scene or to see just what all the fuss is about, point your browser at scene.org and be amazed at the skills and dedication these people have :D

If you come up with a cool effect of your own, post it on the code snippets board for others to use and learn from.

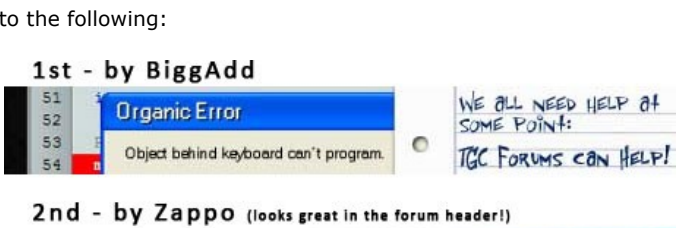
Till next month, Cya.

Download the Full Scene Effects Package, inc. EXE versions (28 MB)
or Download the Source Only Scene Effects Package (14 MB)

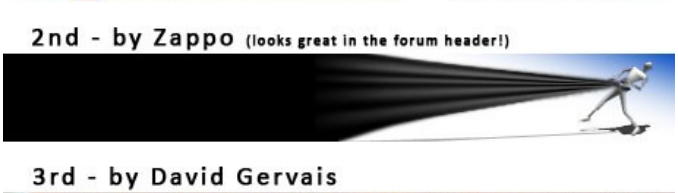
The Forum Banner Winners

We've been running a mini compo on our forum to design new banners to go into rotation in the forum header. Lots of you entered! and some of the banners were excellent, with some real works of pixel art in there, along with all the 10 seconds in Windows Paint efforts :) I put all of the short-listed banners up to a public vote, over 300 of you voted, and here are the results. Congratulations to the following:

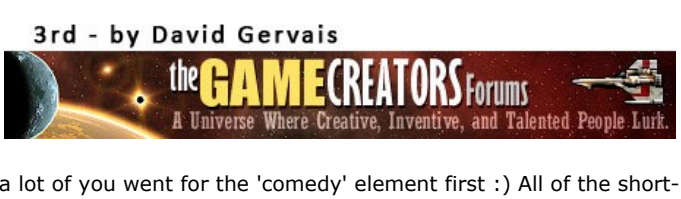
1st - by BiggAdd


51
52
53
54
Organic Error
Object behind keyboard can't program.
We all need help at some point!
TGC Forums can help!

2nd - by Zappo (looks great in the forum header!)



3rd - by David Gervais


the GAME CREATORS Forums
A Universe Where Creative, Inventive, and Talented People Lurk.

It would appear a lot of you went for the 'comedy' element first :) All of the short-listed banners will appear in rotation on the site after the winners have had a bit of the lime-light for a week or two. Would the winners please email me to claim your prizes, thanks.

New Matrix1Utils plugins collection

We received news from IanM about this awesome new release just today...

"I'm releasing 14 new plugins, with ini files and help for you all to use in whatever way you choose:"

- 01 * memory banks - memblocks, with a few new options added
- 02 * basic timing - metronome type ticking, and hiresolution timing
- 03 * mutexes - system level synchronisation between processes (primarily for use with shared-memory banks)
- 04 - raw memory & string allocation - low level stuff only of use to masochists
- 05 - find free ... - locate free images, objects ... everything really
- 06 - Error message control - Take a little control over DBPro error messages, or even trigger your own (with a line number!)
- 07 * General functions - min/max, plus a few numeric functions that are missing from DBPro
- 08 * Random numbers
- 09 - Extra matrix and camera commands
- 10 - Logging - Output logging info from your app, in nicely formatted html or rtf if you want to.
- 13 * Desktop & window info + manipulation
- 15 - Extra WAIT commands - including system-friendly versions that use no system time
- 16 * Strings - One or two string functions that you might find useful
- 17 - 64 bit bitset manipulation - DBPro binary functions don't work with double integers, these functions do

"Yes, I know that I've skipped a few numbers here and there - they'll come along soon enough. All the ones marked with an asterix will have new expanded versions released soon too ... hopefully." They're all in a zip file for easy installation - just let winzip extract it to your DBPro installation folder and it'll all go into the right place.

Read more and download the plugins from this forum thread.

Synergy IDE Update

Synergy IDE has been updated to version 1.0.3, bringing with it a vast array of fixes, tweaks and additions. Synergy IDE now has integrated offsite storage and an improved local backup engine. Teams can now use the Offsite Storage System to share projects without the need of emailing or instant message, just upload your source, and it is instantly available to others. Synergy IDE is now even more streamlined with, a faster start up time, a cleaner interface and less dialogs.

For more information visit this forum thread, and to purchase visit http://www.digitalzenith.net/SynergyIDE.asp

This months Winner

Each month we pluck the email address of one lucky subscriber from the list, and award them with a free copy of DarkBASIC Professional. To claim it all you have to do is send us an email (from the winning email address, naturally) The winner this month is: mesquita@???????????.pt

Outro

Share your news with over 12,500 active games developers!

If you have something you'd like featured in a future issue then please get in touch and include as much information as possible (including where applicable: screen shots, URLs and zip files).
Issue 44 deadline - August 30th.

Freax quote: "And when the last 6510 chip on earth lets out its magical smoke the C-64 coder peering at the darkened screen will sigh: 'What a nice time it was ...'"

The Game Creators in no way endorse or support products mentioned in this newsletter that are not specifically published by them. Always check first before parting with your money, we cannot be held responsible for orders placed with 3rd party companies.

© Copyright The Game Creators Limited 2006. No part of this newsletter may be reproduced without permission.